

FIG. 2

200

```
foreach instruction {add mpy sub} {  
  # for each processor  
  foreach p {s p} {  
    # for each conditional execution  
    foreach ce {e t. f.} {  
      # for each unit that the instruction holds  
      foreach unit {a m d} {  
        # foreach testvector  
        foreach columns in answerset {  
          # generate test with these parameters  
        }  
      }  
    }  
  }  
}
```

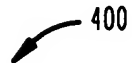
FIG. 3

300

| | |
|------------------------------------|---|
| set instruction(MPY,FORMAT,1sw) | {RTE RX RY} |
| set instruction(MPY,FORMAT,1uw) | {RTE RX RY} |
| set instruction(MPY,FORMAT,2sh) | {RTE RX RY} |
| set instruction(MPY,FORMAT,2uh) | {RTE RX RY} |
| set instruction(MPY,RFACCESS) | {{WRITE} {READ} {READ}} |
| set instruction(MPY,DATATYPES) | {1sw 1uw 2sh 2uh} |
| set instruction(MPY,DIFFDATATYPES) | {{1sw 1sd 1sw 1sw}{1uw 1ud 1uw 1uw}\ {2sh 2sw 2sh 2sh}{2uh 2uw 2uh 2uh}} |
| set instruction(MPY,PROCS) | {s p} |
| set instruction(MPY,UNITS) | {m} |
| set instruction(MPY,CE) | {e t.f.c n v z} |
| set instruction(MPY,CC) | {e} |
| set instruction(MPY,COMBO) | {e} |
| set instruction(MPY,SUFFIX) | {e} |
| set instruction(MPY,CYCLES) | 2 |

FIG. 4

400



```
#setting up state
set instruction(MPY,AS,RXb)  {{maxint}          {minint}          }
set instruction(MPY,AS,RYb)  {{maxint}          {minint}          }
set instruction(MPY,AS,Cb)   {{0}              {0}              }
set instruction(MPY,AS,Vb)   {{0}              {0}              }
set instruction(MPY,AS,Nb)   {{0}              {0}              }
set instruction(MPY,AS,Zb)   {{0}              {0}              }

#specifying desired state.
set instruction(MPY,AS,RTa)  {{mpexpr[maxint]*[maxint]} {mpexpr[minint]*[minint]} }
set instruction(MPY,AS,Ca)   {{0}              {0}              }
set instruction(MPY,AS,Va)   {{0}              {0}              }
set instruction(MPY,AS,Na)   {{sign0unsil}        {0}              }
set instruction(MPY,AS,Za)   {{0}              {sign0unsil}      }
```